

We Claim:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

1. In a graphics system including graphics circuits coupled to an embedded frame buffer, an anti-aliasing method comprising:

(a) rendering a multisampled data representation in the embedded frame buffer;

(b) storing the rendered multisampled data representation in the embedded frame buffer; and

(c) resampling the embedded frame buffer contents to provide an anti-aliased image.

2. The method of claim 1, further including defining a sample pattern for use in rendering the multisampled data representation, and using a reconstruction filter during resampling of the embedded frame buffer, wherein the reconstruction filter uses multisamples from more than one pixel region to obtain data for a resulting pixel.

3. The method of claim 2, wherein a particular support area for the reconstruction filter is determined based on the sample pattern.

4. The method of claim 1, further including varying a sample pattern for multisamples among adjacent pixels, and using a reconstruction filter during resampling having a support region that extends beyond a single pixel

5. The method of claim 4, further including defining a particular support region for the reconstruction filter based on a particular sample pattern for the multisamples.

1 6. In a graphics system of the type that generates an image comprising
2 plural pixels, an anti-aliasing method comprising:

3 (a) generating a multisampled data representation of an image having
4 plural samples associated with each of the plural pixels; and

5 (b) resampling the multisampled data representation, wherein the
6 resampling includes blending at least one of the plural samples into plural image
7 pixels.

1 7. The method of claim 6, further including storing the multisampled data
2 representation in an embedded frame buffer, and further wherein the resampling
3 includes resampling from the embedded frame buffer.

1 8. The method of claim 6, further including a sampling pattern having a
2 non-uniform spatial distribution for the plural samples within neighboring pixels.

1 9. The method of claim 7, further including using a blending filter for the
2 blending which has a support region that is greater than a single pixel.

1 10. The method of claim 8, further including using a blending filter for the
2 blending which has a support region that is greater than a single pixel and is
3 defined based on the sampling pattern.

1 11. The method of claim 10, wherein the support region covers a current
2 pixel and at least a portion of at least two neighboring pixels to the current pixel.

1 12. An anti-aliasing method, comprising:
2 (a) providing plural supersamples within each pixel of a pixel array;
3 (b) varying the spatial distribution of the supersamples within
4 neighboring pixels of the pixel array;

(c) applying, to the array, an anti-aliasing filter having a pixel aperture including a current pixel and at least one of the supersamples from at least two neighboring pixels to the current pixel; and

further including storing the pixel array in an embedded frame buffer, and applying the anti-aliasing filter during a copy out operation from the embedded frame buffer to an external destination.

13. The method of claim 12, wherein the varying of the supersamples defines a sample pattern, and further including defining the aperture of the antialiasing filter based on the sample pattern

14. The method of claim 13, wherein the sample pattern repeats on a pixel quad basis.

15. The method of claim 14, wherein the sample pattern is different for each pixel in a pixel quad.

16. In a graphics chip including an embedded frame buffer, an anti-aliasing method comprising:

- (a) storing a supersampled image in the embedded frame buffer;
- (b) transferring the stored image from the embedded frame buffer to an off-chip destination; and
- (c) in the process of transferring the image, resampling the image to provide an anti-aliased version of the image.

17. The method of claim 16, further including defining a sampling pattern for use in generating the supersampled image, wherein the sampling pattern varies between adjacent pixels of the image.

1 18. The method of claim 17, wherein the resampling includes using a
2 blending filter having a pixel aperture which is greater than one pixel.

1 19. The method of claim 18, further including defining the pixel aperture
2 based on the sampling pattern.

1 20. In a graphics system, a method of anti-aliasing super-sampled pixels,
2 comprising the steps of:

3 (a) defining, within an embedded frame buffer, super-sample locations for
4 each of a plurality of neighboring pixels of an image;

5 (b) assigning color data to each of said super-sample locations; and

6 (c) blending color data from at least two samples obtained from locations
7 defined in step (a) to provide a pixel final color value.

1 21. The method of claim 20, wherein the defining step (a) comprises
2 programming variable sample locations.

1 22. The method as in claim 20, wherein the defining step (a) comprises
2 defining three sample locations for each pixel in a 2 X 2 pixel quad.

1 23. The method as in claim 20, further including wherein the defining step
2 (a) comprises programming sample locations as x and y distances in units of one-
3 twelfth of a pixel.

1 24. The method as in claim 20, further including using a coverage mask to
2 enable/disable samples corresponding to locations defined in step (a), the coverage
3 mask being based at least in part on corresponding portions of each pixel that are
4 occupied by a primitive fragment; and wherein the coverage mask is further based
5 on depth comparisons of primitive fragments at the sample locations.

1 25. The method as in claim 24, wherein the coverage mask comprises a
2 masking bit corresponding to each sample location in a quad of pixels.

1 26. The method as in claim 20, wherein color data associated with pixels
2 is stored within a random access memory embedded within a graphics chip, and
3 step (c) is performed during an operation of transferring data from the embedded
4 random access memory to a memory external of the graphics chip.

1 27. The method of claim 20, wherein the blending step (c) includes
2 assigning blending weights to one or more samples, and blending enabled samples
3 based at least in part on assigned weights.
4

1 28. The method of claim 27, wherein the weights are assigned via an API
2 program function.

1 29. The method of claim 28, wherein the weights are defined in multiples
2 of 1/64.

1 30. The method of claim 20, wherein the blending step (c) includes
2 assigning weights for seven of the samples, and blending the seven samples based
3 at least in part on assigned weights.

1 31. The method of claim 30, wherein the weights are assigned via an API
2 program function.

1 32. The method of claim 20, wherein the blending step (c) includes
2 blending seven samples including three samples from a current pixel with two
3 samples taken from a pixel immediately above the current pixel and two samples
4 taken from a pixel immediately below the current pixel.

1 33. The method of claim 20, wherein each super-sampled pixel is
 2 represented in memory by at least three samples of 16-bit color data and three
 3 samples of corresponding 16-bit Z position data.

1 34. A graphics system, an apparatus for anti-aliasing super-sampled
 2 pixels, comprising:
 3

4 means for programmably defining three sample locations for obtaining
 5 super-sampled color data associated with a pixel for each of a plurality of
 6 neighboring pixels;

7 coverage mask means to enable/disable samples corresponding to said
 8 sample locations, the coverage mask means being based at least in part on
 9 corresponding portions of each pixel that are occupied by rendered primitive
 10 fragments; and

11 color data blending filter means for combining color data from at least two
 12 super-sampled color data to provide a pixel final color value.

1 35. The system of claim 34, wherein said blending filter means comprises
 2 a means for computing a weighted average of samples.

1 36. The system of claim 34, wherein said blending filter means comprises
 2 a means for computing a weighted average of color data of at least three samples
 3 corresponding to a current pixel and at least two samples corresponding to a pixel
 4 immediately above the current pixel and at least two samples corresponding to a
 5 pixel immediately below the current pixel.

1 37. The system of claim 34, wherein the blending filter means further
 2 comprises a weighting coefficient means for selectively weighting each sample of
 3 color data for computing a weighted average of color data, the graphics system

including a means for programmably defining a weight coefficient associated with each sample.

38. In a graphics system, a method of providing full-scene anti-aliasing, comprising the steps of:

(a) defining three super-sampled color data locations associated with a pixel for each of a plurality of neighboring pixels;

(b) blending the three super-sampled color data locations within two super-sampled color locations of a pixel immediately above the current pixel and two super-sampled color locations of a pixel immediately below the current pixel; and

(c) displaying a pixel having a color corresponding to the blend.

39. The method of claim 38, wherein the blending step (b) includes assigning weights for the seven super-sampled color data locations, and computing a weighted average based at least in part on assigned weights.

40. In a graphics system, a method of anti-aliasing pixels wherein each pixel is subdivided into a plurality of super-sampled portions identified by locations programmably defined therein, comprising the steps of:

(a) defining a plurality of super-sampled locations for each of a plurality of neighboring pixels;

(b) using coverage masks to develop color data for super-samples corresponding to locations defined in step (a), the coverage masks being based at least in part on corresponding portions of each pixel that are occupied by primitive fragments; and

(c) blending color data from at least two selected super-samples obtained from locations defined in step (a) during a copy-out operation to provide a filtered pixel color value.

1 41. In a graphics system, a pixel data processing arrangement having a
2 multi-tap selectable-weight blending filter characterized by a vertically-arranged
3 multiple-pixel filter support region wherein one or more color data samples from a
4 plurality of vertically disposed pixels are blended to form a pixel color.

1 42. In a graphics system, a pixel data processing arrangement for
2 providing full-scene anti-aliasing and/or de-flickering interlaced displays,
3 comprising:

4 a frame buffer containing super-sampled pixel data for a plurality of pixels;
5 a plurality of scan-line buffers connected to receive super-sampled pixel
6 color data from the frame buffer; and

7 a multi-tap selectable-weight blending filter coupled to the scan-line buffers,
8 the blending filter characterized by a vertically-arranged multiple-pixel filter
9 support region wherein one or more color data samples from a plurality of
10 vertically disposed pixels are blended to form a pixel color.

1 43. An apparatus for anti-aliasing as set forth in claim 42, wherein pixel
2 data in the frame buffer also includes depth (Z data) information.

1 44. An arrangement that anti-aliases super-sampled pixels comprising:
2 an embedded frame buffer storing three super-sample locations within each
3 pixel of a pixel array, each said super-sample location having a corresponding
4 color value; and

5 a one-dimensional color data blending filter that blends the three super-
6 sample color values with super-sample color values of adjacent neighboring pixels
7 while information within the embedded frame buffer is being transferred to a
8 destination.

1 45. The arrangement of claim 44, wherein the embedded frame buffer stores
2 no more than three super-sample locations within each pixel.

1 46. The arrangement of claim 44, wherein the filter blends super-sample
2 color values corresponding to each pixel with super-sample color values
3 corresponding to at least one further neighboring pixel.

1 47. The arrangement of claim 44, wherein the filter blends super-sample
2 color values corresponding to three vertically aligned pixels to produce a screen
3 pixel output.

1 48. An anti-aliasing method comprising:
2 programmably defining plural super-sampled locations within at least one
3 screen pixel, each said super-sampled location having a corresponding color value;
4 and

5 blending said super-sampled color values using a vertical filter during a
6 copy-out operation from an embedded frame buffer to an external frame buffer.

1 49. Within a pixel quad having first, second, third and fourth pixels and a
2 quad center, a method of defining an optimal set of three super-sampling locations
3 for anti-aliasing, said method comprising:

4 (a) defining a first set of super-sample locations for a first pixel in the pixel
5 quad at the following coordinates (range 1-12) relative to the quad center:

6 (12,11)

7 (4,7)

8 (8,3);

9 (b) defining a second set of super-sample locations for a second pixel in the
10 pixel quad at the following coordinates (range 1-12) relative to the quad center:

11 (3,11)

12 (11,7)

13 (7,3);

14 (c) defining a third set of super-sample locations for a third pixel in the pixel
15 quad at the following coordinates (range 1-12) relative to the quad center:

16 (2,2)

17 (10,6)

18 (6,10);

19 (d) defining a fourth set of super-sample locations for a fourth pixel in the
20 pixel quad at the following coordinates (range 1-12) relative to the quad center:

21 (9,2)

22 (1,6)

23 (5,6);

24 (e) using a resampling filter having a support area that uses three
25 supersamples from a current pixel, two super-samples from a pixel immediately
26 above the current pixel, and two samples from a pixel immediately below the
27 current pixel; and

28 (e) using respective weighting coefficients in the resampling filter
29 having the following values: $1/12$, $1/6$, $1/6$, $1/6$, $1/6$, $1/6$, $1/12$.